```
***********************************
*  PROCEDURES LISP D'ANALYSE      *
*  DE RELATIONS HARMONIQUES.      *
*                                 *
*  PATRICK GREUSSAY               *
***********************************
```

RESUME :

L'ARTICLE DECRIT UN ENSEMBLE DE FONCTIONS LISP 510 DESTINE A LIVRER, A PARTIR D'UN ENSEMBLE DE HAUTEURS DONNE, LES RELATIONS HARMONIQUES LES PLUS ELEMENTAIRES.
A SAVOIR : DOMINANTE, SOUS-DOMINANTE, TRITON, NAPOLITAINE, SENSIBLE, SOUS FORME PUREMENT SYMBOLIQUE.

L'ARTICLE A ETE CORRECTEMENT COMPILE ET INTERPRETE PAR LE SYSTEME LISP 510.

L'ENSEMBLE DE CES FONCTIONS CONSTITUE UNE ZONE DE SERVICE D'UN PROGRAMME PLUS IMPORTANT D'ANALYSE DE FONCTIONS HARMONIQUES DANS L'OEUVRE DE CLAUDE DEBUSSY.
ETANT DONNE UN ENSEMBLE DE REGIONS HARMONIQUES ET LEURS REPRESENTANTS REPERES, L'EMPLOI CONSTANT DE L'OSCILLATION ENTRE -NOM- ET -VALEUR- DE HAUTEUR SE REVELE FORT IMPORTANT DANS LE TRAVAIL COMPOSITIONNEL DE CE MUSICIEN. TOUTE ANALYSE EN PROFONDEUR SE MEPRENDRAIT A N'EN PAS TENIR COMPTE.

POUR L'AUTOMATISATION (PARTIELLE) DES PROCEDURES D'ANALYSE, LE LANGAGE LISP A ETE CHOISI POUR LA FACILITE QU'IL OFFRE DE RENDRE TOUT OBJET SUSCEPTIBLE DE RECEVOIR DES VALEURS (TANT SUR SA P-LISTE QUE SUR LA A-LISTE).

ON DISTINGUERA POUR CHAQUE HAUTEUR DEUX TYPES DE -NOMS- :
- . NUMERIQUE (MODULO 12)
- . LITERAL

REPRESENTES PAR LES DEUX LISTES DE PROPRIETES -1-L ET -2-L.
CES CORRESPONDANCES ONT ETE CHOISIES POUR TRAITER CONVENABLEMENT LES PHENOMENES DE SYNONYMIE (EX: DO # - RE BEML).

POUR CHAQUE LITERALE ON UTILISERA ALTERNATIVEMENT
- . UN ATOME (DANS LES CAS DE NON ALTERATION)
- . UNE PAIRE POINTEE (ALTERATION . ATOME)

NOTA : '*' REPRESENTE LE BEMOL.

```
(MODE 2)

RPLACA (
        -D-L
        (DO SOL RE LA MI SI FA DO)
)
RPLACA (
        -S-L
        (DO RE MI FA SOL LA SI DO)
)
RPLACA (
        -1-L
        ((0 (DO (# . SI))) (1 ((# . DO) (* . RE)))
         (2 (RE)) (3 ((# . RE) (* . MI))) (4 (MI (* . FA)))
         (5 ((# . MI) FA)) (6 ((# . FA) (* . SOL))) (7 (SOL))
         (8 ((# . SOL) (* . LA))) (9 (LA))
         (10 ((# . LA) (* . SI))) (11 (SI (* . DO))))
)
RPLACA (
        -2-L
        (((# . SI) . 0) (DO . 0) ((# . DO) . 1)
         ((* . RE) . 1) (RE . 2) ((# . RE) . 3) ((* . MI) . 3)
         ((* . FA) . 4) (MI . 4) ((# . MI) . 5) (FA . 5)
         ((# . FA) . 6) ((* . SOL) . 6) (SOL . 7)
         ((# . SOL) . 8) ((* . LA) . 8) (LA . 9)
         ((# . LA) . 10) ((* . SI) . 10) (SI . 11)
         ((* . DO) . 11))
)

DEFINE ((

        ;       ETANT DONNE UNE LITERALE, P-1-L LIVRE L'ENTIER
                DANS (0, 11) CORRESPONDANT DANS LA LISTE -1-L. ;
(P-1-L
   (LAMBDA (X)
       (CDR (SASSOC X -1-L))))

             ; ETANT DONNE UN ENTIER PRIS DANS L'INTERVALLE
               (0, 11), P-2-L LIVRE LA LISTE DES LITERALES
               ASSOCIEES A CET ENTIER DANS LA LISTE -2-L.
                 EN FAIT P-2-L EST UNE GENERALISATION DE
               (SASSOC X L FN) X ETANT UNE S-EXPRESSION
               QUELCONQUE.                                  ;
(P-2-L
   (LAMBDA (X)
       (CDR (OR
                (SASSOC X -2-L)
                (PROG (Y)
                  (SETQ Y -2-L)
             A    (AND
                      (EQUAL X (CAAR Y))
                      (RETURN (CAR Y)))
                  (SETQ Y (CDR Y))
                  (GO A))))))
```

```
(ABS
   (LAMBDA (X)
      (COND
            ((GT X 0) X)
            (T (DIFFER 0 X)))))

      ;        CETTE FONCTION A POUR ARGUMENTS :
                 (NOM1 NOM2 NUM1 NUM2 LISTE-REPERE)
               ELLE VERIFIE SI   . DEUX NOMS SATISFONT
                  A LA RELATION NUMERIQUE:
                         NUM1 = ABS (VAL-NOM1 - VAL-NOM2)
                              . LES DEUX NOMS EN TANT QUE
             COUPLE DE LITERALES FORMENT OU NON UNE SOUS-SUITE
             DE LA LISTE REPERE (PAR MATCHING LITERAL, AU
             MOYEN DE LA FONCTION CSQ).
             (EX: MALGRE LES MEMES VALEURS NUMERIQUES
             LE COUPLE DO# - LA BEML N'EST PAS UNE QUINTE).;
(IS?
   (LAMBDA (X Y M N L Z)
      (PROGN
            (SETQ Z (ABS (DIFFER (P-2-L X) (P-2-L Y))))
            (COND
                 ((OR
                     (EQ Z M)
                     (EQ Z N))
                  (CSQ L X Y)))))))

     ;RAPPELS :
             1 . BIEN QUE GET SOIT EVIDEMMENT UNE SUBR, VOICI
             SA DEFINITION EN S-EXPRESSIONS:
               (GET (LAMBDA (X Y) (COND
                                       ((NULL X))
                                       ((EQ (CAR X) Y)
                                        (CADR X))
                                       (T (GET (CDR X) Y)) )))
             2 . EN LISP 510, SI -TOUS- LES PREDICATS
             D'UN COND S'EVALUENT EN NIL, LA VALEUR DE COND
             EST ALORS NIL, ET NE PROVOQUE PAS DE DIAGNOSTIC
             D'ERREUR.
             3 . UNE FONCTION PEUT COMPORTER A L'APPEL :
               - PLUS D'ARGUMENTS QUE PREVU. ILS SERONT
                 EVALUES MAIS NON TRANSMIS.
               - MOINS QUE PREVU : LES ARGUMENTS ABSENTS
                 SERONT PAR DEFAUT EVALUES A NIL.


              LA FONCTION CSQ , SI LE COUPLE (X1 Y1) EST
             ACCEPTABLE COMME ELEMENT DE LA RELATION SUR LA
             LISTE L, RAMENE EN VALEUR CE COUPLE, CONVE-
             NABLEMENT ORDONNE. ;
```

```
(CSQ
   (LAMBDA (L X1 Y1)
      (PROGN
            (SETQ X1 (COND
                          ((ATOM X1) X1)
                          (T (CDR X1))))
            (SETQ Y1 (COND
                          ((ATOM Y1) Y1)
                          (T (CDR Y1))))
            (COND
                 ((EQ (GET L X1) Y1)
                  (LIST X Y))
                 ((EQ (GET L Y1) X1)
                  (LIST Y X))))))

      ;      ?IS? APPLIQUE LA FONCTION F AUX ARGUMENTS
             X ET Y.;
(?IS?
   (LAMBDA (X Y F)
      (PROGN
            (SETQ X (F X Y))
            (COND
                 (X (LIST (CADR X) (CAR X)))))))


      ;        VOICI LE GROUPE DES FONCTIONS
                 D       QUI                DOMINANTE
                 N       CORRESPONDENT      NAPOLITAINE
             IS?-T       RESPECTIVEMENT     TRITON
                 SD      AUX                SOUS-DOMINANTE
                 S       PREDICATS :        SENSIBLE ;
(IS?-D
   (LAMBDA (X Y)
      (IS? X Y 5 7 -D-L)))
(IS?-N
   (LAMBDA (X Y)
      (?IS? X Y (QUOTE IS?-S))))
(IS?-T
   (LAMBDA (X Y)
      (COND
           ((EQ 6 (ABS (DIFFER (P-2-L X) (P-2-L Y))))
            (LIST X Y)))))
(IS?-SD
   (LAMBDA (X Y)
      (?IS? X Y (QUOTE IS?-D))))
(IS?-S
   (LAMBDA (X Y)
      (IS? X Y 1 11 -S-L)))
```

```
(X-$REL
   (LAMBDA (F NSH Y Z)
      (PROGN
            (SETQ Z (F (CAR M) (CAR N)))
            (AND
               Z
               (SET NSH (CONS Z Y))))))
($RELATIONS
   (LAMBDA (L)
      (PROG (M N L-D L-SD L-T L-N L-S)
             (TERPRI)
             (SETQ M L)
      A      (AND
                (NULL (CDR M))
                (RETURN (EDITL)))
             (SETQ N (CDR M))
      B      (X-$REL (QUOTE IS?-D)  (QUOTE L-D) L-D)
             (X-$REL (QUOTE IS?-SD) (QUOTE L-SD) L-SD)
             (X-$REL (QUOTE IS?-N)  (QUOTE L-N) L-N)
             (X-$REL (QUOTE IS?-T)  (QUOTE L-T) L-T)
             (X-$REL (QUOTE IS?-S)  (QUOTE L-S) L-S)
             (AND
                (SETQ N (CDR N)) (GO B))
             (SETQ M (CDR M))
             (GO A))))
(EDITL
   (LAMBDA ()
      (PROGN
            (PRINT (QUOTE L-D))
            (PRINT L-D)
            (PRINT (QUOTE L-SD))
            (PRINT L-SD)
            (PRINT (QUOTE L-T))
            (PRINT L-T)
            (PRINT (QUOTE L-N))
            (PRINT L-N)
            (PRINT (QUOTE L-S))
            (PRINT L-S))))
))
```

```
$RELATIONS ((DO RE MI FA SOL LA SI))

L-D
( ( MI SI ) ( LA MI ) ( RE LA ) ( SOL RE ) ( DO SOL ) ( FA DO
) )
L-SD
( ( SI MI ) ( MI LA ) ( LA RE ) ( RE SOL ) ( SOL DO ) ( DO FA
) )
L-T
( ( FA SI ) )
L-N
( ( FA MI ) ( DO SI ) )
L-S
( ( MI FA ) ( SI DO ) )
```

```
          ; * * DEBUSSY : NUAGES , MES 1 ;

$RELATIONS ((SI (# , FA) (# , DO) MI (* , SI) RE (# , LA) LA))

L-D
( ( RE LA ) ( LA MI ) ( ( # , FA ) ( # , DO ) ) ( MI SI ) (
SI ( # , FA ) ) )
L-SD
( ( LA RE ) ( MI LA ) ( ( # , DO ) ( # , FA ) ) ( SI MI ) ( (
# , FA ) SI ) )
L-T
( ( MI ( # , LA ) ) ( MI ( * , SI ) ) )
L-N
( ( ( * , SI ) LA ) ( RE ( # , DO ) ) ( SI ( # , LA ) ) )
L-S
( ( LA ( * , SI ) ) ( ( # , DO ) RE ) ( ( # , LA ) SI ) )
```

NOTE :

MALGRE SON *EXTREME* SIMPLICITE, JE SUIS TOUT A FAIT CONSCIENT DE LA DIFFICULTE DE LECTURE DE CE PROGRAMME POUR LES PERSONNES PEU FAMILIERES AVEC LE LANGAGE LISP.

IL EST CLAIR QUE CETTE DIFFICULTE EST CHRONIQUE.

RESTE QU'APRES UN CERTAIN TEMPS D'ACCOUTUMANCE, L'ELEGANCE ET LA CONCISION DE CE LANGAGE DEVIENNENT FRAPPANTES ET PERMETTENT LA NOTATION ET L'EXECUTION DE PROCESSUS FORT LOURDS A EXPRIMER AUTREMENT.

DE PLUS, LE PROGRAMMEUR D'EXPERIENCE Y RECONNAITRA A COUP SUR LA FACILITE BIEN CONNUE QUI LUI FAIT SOUVENT PREFERER UN LANGAGE MACHINE COMME SUPPORT D'EXPRESSION DE SES ALGORITHMES, A SAVOIR L'ACCES DYNAMIQUE A TOUS LES OBJETS, EN PARTICULIER POINTEURS (TRAIT QUE PL/1 POSSEDE EN PARTIE) ET DONNEES EXECUTABLES I.E. PROGRAMMES.

ENFIN, EN CE QUI CONCERNE LES MUSICIENS, ILS PRENNENT A CE LANGAGE UN PIED CERTAIN.